WHAT IS CLAIMED IS:

1. A microprocessor having a branch prediction that the branch will be approved,

employing a limited conditional branch instruction whose instruction to be executed next when the branch prediction is not hit is limited;

wherein, if it is detected that the branch is not approved in decode stage for the limited conditional branch instruction, a fetch stage and a decode stage for the next instruction to be executed are conducted in fewer machine cycles than required for a fetch stage and a decode stage for a normal instruction.

2. A microprocessor having a branch prediction that the branch will be approved,

employing a limited conditional branch instruction whose next instruction to be executed next when the branch prediction is not hit is limited, comprising:

a first memory for storing instructions;

a second memory for storing an op code of the instruction to be executed next when the branch prediction is not hit;

wherein, if it is detected that the branch is not approved in the decode stage for the limited conditional branch instruction, an op code is provided from the second memory to the decoder and an operand is provided from the first memory to the decoder.

3. A microprocessor having a branch prediction that the branch will be approved,

employing a limited conditional branch instruction whose instruction to be executed next when the branch prediction is not hit is limited, comprising:

a first decoder, as a basic decoder, used for decoding a normal instruction; and

a second decoder, as a dedicated decoder, used for decoding an instruction to be executed next when the branch of the limited conditional branch instruction is not approved, wherein the instruction to be executed next is decoded within fewer machine cycles than required for a normal instruction;

wherein when it is detected that the branch is not approved by the first decoder in decoding the limited conditional branch instruction, the second decoder is used for the decode stage for the instruction to be executed next.

4. An instruction converter;

employing a limited conditional branch instruction whose instruction to be executed next when the branch prediction is not hit is limited;

wherein when detecting the conditional branch instruction in the inputted instruction sequence, checking the instruction to be executed next when the conditional

branch is not approved, and checking whether the relationship between the conditional branch instruction and the instruction to be executed next when the branch is not approved corresponds to the relationship between the limited branch instruction and the instruction to be executed next when the branch is not approved, and if it corresponds, converting the conditional branch instruction to the limited conditional branch instruction.

5. A microprocessor having a branch prediction that the branch will not be approved, employing a limited conditional branch instruction whose branch designation instruction to be executed next when the branch prediction is hit is limited;

wherein, if it is detected that the branch is approved in decode stage for the limited conditional branch instruction, a fetch stage and a decode stage for a branch designation instruction to be executed next are conducted within fewer machine cycles than required for a fetch stage and a decode stage for a normal instruction.

6. A microprocessor having a branch prediction that the branch will not be approved, employing a limited conditional branch instruction whose branch designation instruction to be executed next when the branch prediction is hit is limited, comprising:

a first memory for storing instructions;

a second memory for storing op code of the branch designation instruction to be executed next when the branch prediction is hit;

wherein, if it is detected that the branch is approved in the decode stage for the limited conditional branch instruction, an op code is provided from the second memory to the decoder and an operand is provided from the first memory to the decoder.

7. A microprocessor having a branch prediction that the branch will not be approved, employing a limited conditional branch instruction whose branch designation instruction to be executed next when the branch prediction is hit is limited, comprising:

a first decoder, as a basic decoder, used for decoding a normal instruction; and

a second decoder, as a dedicated decoder, used for decoding a branch designation instruction to be executed next when the branch is approved for the limited conditional branch instruction, wherein the branch designation instruction is conducted within fewer machine cycles than required for a normal instruction;

wherein when it is detected that the branch is approved by the first decoder in decoding the limited conditional branch instruction, the second decoder is used for the decode stage for the branch designation instruction to be executed next.

8. An instruction converter;

employing a limited conditional branch instruction whose branch designation instruction to be executed next when the branch prediction is hit is limited;

wherein when detecting the conditional branch instruction in the inputted instruction sequence, checking the branch designation instruction to be executed next when the conditional branch is approved, and checking whether the relationship between the conditional branch instruction and the branch designation instruction corresponds to the relationship between the limited branch instruction and the branch designation instruction, and if it corresponds, converting the conditional branch instruction to the limited conditional branch instruction.

9. A microprocessor employing a limited unconditional branch instruction whose branch designation instruction to be executed next is limited,

wherein, if the limited unconditional branch instruction is detected in a decode stage, a fetch stage and a decode stage for the branch designation instruction of the limited unconditional branch instruction are conducted within fewer machine cycles than required for a fetch stage and a decode stage for a normal instruction.

10. A microprocessor employing a limited unconditional branch instruction whose branch designation instruction to be executed next is limited, comprising:

a first memory for storing instructions;

a second memory for storing an op code of the branch designation instruction of the limited unconditional branch instruction;

wherein, if the limited unconditional branch instruction is detected in a decode stage, the op code is provided from the second memory to the decoder quickly and the operand is provided from the first memory to the decoder quickly.

11. A microprocessor employing a limited unconditional branch instruction whose branch designation instruction to be executed next is limited, comprising:

a first decoder used for decoding a normal instruction; and

a second decoder used for decoding a branch designation instruction of the limited unconditional branch instruction;

wherein the branch designation instruction is decoded quickly within fewer machine cycles than required for a normal instruction, and when the limited unconditional branch instruction is detected by the first decoder, the second decoder is used for the decode stage for the branch designation instruction of the limited unconditional branch instruction.

12. An instruction converter employing a limited unconditional branch instruction whose branch designation instruction to be executed next is limited,

wherein when detecting the unconditional branch instruction in an inputted instruction sequence, the instruction converter checks the instruction to be executed subsequent to the unconditional branch instruction, and checks whether the relationship between the branch designation instruction and the unconditional branch instruction corresponds to the relationship between the branch designation instruction and the limited unconditional branch instruction, and if it corresponds, converts the unconditional branch instruction to the limited unconditional branch instruction.

13. A method for driving a microprocessor having a branch prediction that a branch will be approved,

the microprocessor employing a limited conditional branch instruction whose instruction to be executed next when the branch prediction is not hit is limited;

wherein, if it is detected that the branch is not approved in decode stage for the limited conditional branch instruction, a fetch stage and a decode stage for the next instruction to be executed are conducted in fewer machine cycles than required for a fetch stage and a decode stage for a normal instruction.

14. A method for driving a microprocessor having a branch prediction that a branch will be approved,

the microprocessor employing a limited conditional branch instruction whose next instruction to be executed next when the branch prediction is not hit is limited, comprising:

storing instructions into a first memory; and

storing an op code of the instruction to be executed next when the branch prediction is not hit into a second memory;

wherein, if it is detected that the branch is not approved in a decode stage for the limited conditional branch instruction, an op code is provided from the second memory to a decoder and an operand is provided from the first memory to the decoder.

15. A method for driving a microprocessor having a branch prediction that a branch will be approved,

the microprocessor employing a limited conditional branch instruction whose instruction to be executed next when the branch prediction is not hit is limited, comprising:

in case of decoding a normal instruction, using a first decoder as a basic decoder;

and

in case of decoding an instruction to be executed next when the branch of the limited conditional branch instruction is not approved, using a second decoder as a dedicated decoder, wherein the instruction to be executed next is decoded within fewer machine cycles than required for a normal instruction ;

wherein when it is detected that the branch is not approved by the first decoder in decoding the limited conditional branch instruction, the second decoder is used for the decode stage for the instruction to be executed next.

16. A method for converting instructions for a microprocessor that

employs a limited conditional branch instruction whose instruction to be executed next when the branch prediction is not hit is limited;

wherein when a conditional branch instruction in an inputted instruction sequence is detected, the instruction to be executed next when the conditional branch is not approved is checked, and the relationship between the conditional branch instruction and the instruction to be executed next when the branch is not approved is checked to determine whether the relationship corresponds to the relationship between the limited branch instruction and the instruction to be executed next when the branch is not approved, and if it corresponds, converting the conditional branch instruction to the limited conditional branch instruction.

17. A method for driving a microprocessor having a branch prediction that a branch will not be approved,

the microprocessor employing a limited conditional branch instruction whose branch designation instruction to be executed next when the branch prediction is hit is limited;

wherein, if it is detected that the branch is approved in a decode stage for the limited conditional branch instruction, a fetch stage and a decode stage for a branch designation instruction to be executed next are conducted in fewer machine cycles than required for a fetch stage and a decode stage for a normal instruction.

18. A method for driving a microprocessor having a branch prediction that a branch will not be approved,

the microprocessor employing a limited conditional branch instruction whose branch designation instruction to be executed next when the branch prediction is hit is limited, comprising:

storing instructions in a first memory; and

storing an op code of the branch designation instruction to be executed next when the branch prediction is hit into a second memory;

wherein, if it is detected that the branch is approved in a decode stage for the limited conditional branch instruction, an op code is provided from the second memory to the decoder and an operand is provided from the first memory to the decoder.

19. A method for driving a microprocessor having a branch prediction that a branch will not be approved,

the microprocessor employing a limited conditional branch instruction whose branch designation instruction to be executed next when the branch prediction is hit is limited, comprising:

in case of decoding a normal instruction, using a first decoder as a basic decoder; and

in case of decoding a branch designation instruction to be executed next when the branch is approved for the limited conditional branch instruction, using a second decoder as a dedicated decoder, wherein the branch designation instruction is conducted within fewer machine cycles than required for a normal instruction;

wherein when it is detected that the branch is approved by the first decoder in decoding the limited conditional branch instruction, the second decoder is used for the decode stage for the branch designation instruction to be executed next.

20. A method for converting instructions for a microprocessor that

employs a limited conditional branch instruction whose branch designation instruction to be executed next when the branch prediction is hit is limited;

wherein when a conditional branch instruction in an inputted instruction sequence is detected, the branch designation instruction to be executed next when the conditional branch is approved is checked, and the relationship between the conditional branch instruction and the branch designation instruction is checked to determine whether the relationship corresponds to the relationship between the limited branch instruction and the branch designation instruction, and if it corresponds, converting the conditional branch instruction to the limited conditional branch instruction.

21. A method for driving a microprocessor employing a limited unconditional branch instruction whose branch designation instruction to be executed next is limited,

wherein, if the limited unconditional branch instruction is detected in a decode stage, a fetch stage and a decode stage for the branch designation instruction of the limited unconditional branch instruction are conducted within fewer machine cycles than

required for a fetch stage and a decode stage for a normal instruction.

22. A method for driving a microprocessor employing a limited unconditional branch instruction whose branch designation instruction to be executed next is limited, comprising:

storing instructions in a first memory; and

storing an op code of the branch designation instruction of the limited unconditional branch instruction in a second memory;

wherein, if the limited unconditional branch instruction is detected in the decode stage, the op code is provided from the second memory to the decoder quickly and the operand is provided from the first memory to the decoder quickly.

23. A method for driving a microprocessor employing a limited unconditional branch instruction whose branch designation instruction to be executed next is limited, comprising:

in case of decoding a normal instruction, using a first decoder; and

in case of decoding a branch designation instruction of the limited unconditional branch instruction, using a second decoder;

wherein the branch designation instruction is decoded quickly, within fewer machine cycles than required for a normal instruction, when the limited unconditional branch instruction is detected by the first decoder, the second decoder is used for a decode stage for the branch designation instruction of the limited unconditional branch instruction.

24. A method for converting instructions for a microprocessor that

employs a limited unconditional branch instruction whose branch designation instruction to be executed next is limited,

wherein when an unconditional branch instruction in an inputted instruction sequence is detected, the instruction converter checks the instruction to be executed subsequent to the unconditional branch instruction, and checks whether the relationship between the branch designation instruction and the unconditional branch instruction corresponds to the relationship between the branch designation instruction and the limited unconditional branch instruction, and if it corresponds, converts the unconditional branch instruction to the limited unconditional branch instruction.